22 EVOLUTION OF SECONDARY SOFTWARE BUSINESSES: Understanding Industry Dynamics

Pasi Tyrväinen
University of Jyväskylä
Jyväskyla, Finland

Juhani Warsta
University of Oulu
Oulu, Finland

Veikko Seppänen
University of Oulu
Oulu, Finland

Abstract *Primary software industry originates from IBM's decision to unbundle software-related computer system development activities to external partners. This kind of outsourcing from an enterprise internal software development activity is a common means to start a new software business serving a vertical software market. It combines knowledge of the vertical market process with competence in software development. In this research, we present and analyze the key figures of the Finnish secondary software industry, in order to quantify its interaction with the primary software industry during the period of 2000–2003. On the basis of the empirical data, we present a model for evolution of a secondary software business, which makes explicit the industry dynamics. It represents the shift from internal software developed for competitive advantage to development of products supporting standard business processes on top of standardized technologies. We also discuss the implications for software business strategies in each phase.*

Keywords Software business, industry development

Please use the following format when citing this chapter:

Tyrväinen, P., Warsta, J., and Seppänen, V., 2008, in IFIP International Federation for Information Processing, Volume 287, Open IT-Based Innovation: Moving Towards Cooperative IT Transfer and Knowledge Diffusion, eds. León, G., Bernardos, A., Casar, J., Kautz, K., and DeGross, J. (Boston: Springer), pp. 381-401.

1 INTRODUCTION

The origins of the software industry (SWI) can be tracked down to the decision by IBM to unbundle some secondary parts of its computer development to independent software companies (Campbell-Kelly 2004). Early on, computing software development was considered a secondary industry serving the core business, which was computer hardware manufacturing. Actually, this is still visible in the brand names of both the global information and communication technology (ICT) giants, such as IBM, and the major professional and scientific establishments of the field, such as ACM.

Gradually, however, the software industry has established a position as an independent industry and as a specialized focus of scientific research interests. Software business related research targeted first the core of the emerging industry, the companies developing and selling independent software products and software services. These companies are often referred to as the *primary software industry* (BMBF 2000). The term *secondary software industry* refers to software businesses being performed by companies focusing on some other industry, but using software as a part of their products or services. Simplifying, the automative industry in the 1970s, the electronics industry in the 1980s, and the telecommunications industry in the 1990s represent, globally, key host industries for secondary software businesses. At the moment, for example, the automotive and to some extent the aerospace industries are in the middle of booming secondary software businesses.

Contemporary research on software business models extensively targets the primary software market, while the business models and evolution of the secondary software market has gained only little attention, even though the importance and dependence on software applications in many fields has been growing. This is unfortunate thinking in that the emergence of new primary software companies is neither limited to new companies nor has the trend to move software businesses from the secondary to the primary industry stopped. Therefore, questions regarding the extent to which secondary software companies shift to primary software businesses, the industries from which they came, and why and how this happens are still valid.

Figure 1 represents the overall setting. The primary software companies (on the left) runs software business as their main activity, but may also include units with other business orientations. These companies are categorized mostly under category 72.2 in the NACE categorization used by OECD and EITO. Companies with other categorizations often have software-oriented business units or their operations include software development, sales, and consulting related to their products. These companies are represented on the right-hand side as the secondary software industry. The official statistics of software production include the total sales of the 72.2 category software companies including non-software sales, while software sales of the software business units in the secondary software industry are excluded.

For example, based on these numbers, according to the OECD, the sales of the Finnish primary software industry was 2,356 M€ out of the total 3,499 M€ of computer and related activities in 2000, the beginning of the observed period (Colecchia et al. 2002), while the consumption of IT products in Finland was 5,070 M€ in the same year (EITO 2004). Estimates of employer volumes have varied as well. Based on estimates

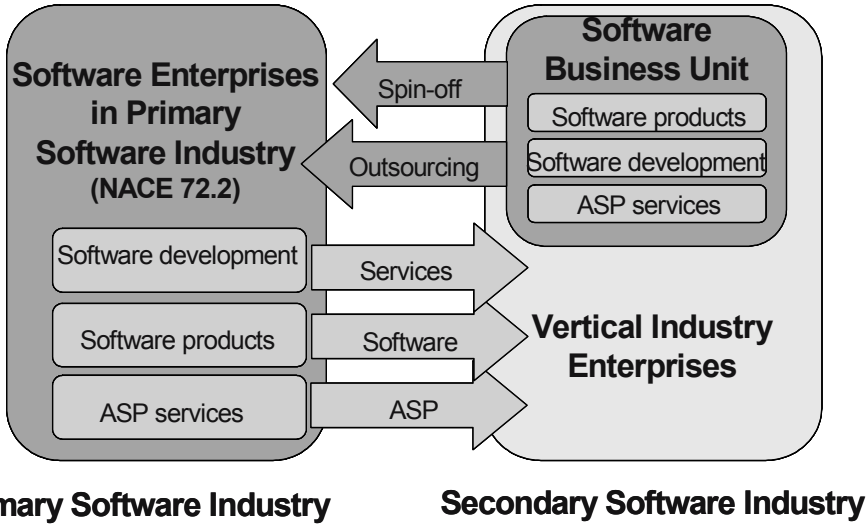

Figure 1. Interaction between Enterprises in the Primary Software and Other Industries, Including the Secondary Software Industry

by representatives of the employees union,[1] software services companies employed about 35,000 people working in software design and implementation related activities in 2003. Of these, about 25,000 were programmers and designers, with an estimated 10,000 others working on different software-related activities. Over 80 percent of the companies employed fewer than five people, indicating that the small companies with little revenue are not visible in all estimates. According to the estimates of an annual software product business review (Kuitunen et al. 2005), the software product business portion of the primary SWI was estimated to grow from approximately 800 M€ to close to a million during the period of observation. Most of these statistics recorded sales based on the location of the headquarters and the primary industry category, regardless of the location of the software business unit and the secondary products and businesses.

According to the European Information Technology Observatory (EITO 2004), out of the 200,000 M€ worth of software products sold annually throughout the world, roughly one half are applications, one third are infrastructure software, and less than one fifth are development tools. Furthermore, out of the applications, close to one half are software meant for some vertical market and close to the second half are ERP and other horizontal software, while less than a tenth of software products were sold to private consumers (Colecchia et al. 2002).

The share of the private consumer market is also extremely small from the over 400,000 M€ software service market, while strong vertical industries drive the direction

[1]From interviews conducted in 2003 with J. Reinsialo, head of the association "Tietoalan toimihenkilöt," whose members are unions representing employees of professional software services in Finland.

of the development. For example, out of its over 400,000 M€ size ICT budget, the finance industry uses 160,000 M€ for software services and 20 M€ for software products, which is more than three times the volume of consumer market sales. To sum it up, the main volume of the primary software industry comes from vertical businesses.

2 FACTORS AFFECTING MARKET AND BUSINESS CHANGE

Previous research on software ventures shows that new primary software companies are born both from academic environments and technology developers, as well as from the business units of secondary software enterprises (Chesbrough and Rosenbloom 2002; Giarratana 2004; Romijn and Albaladejo 2002). Regarding the latter, which is the focus of this research, both establishment of spin-off companies and outsourcing of software development tasks move personnel and business activities to the primary software industry from vertical industries. The initial impact of vertical markets is visible in many generic and horizontal software products. For example, manufacturing resource planning systems developed initially as tailored solutions extended their scope from the first adopters to other functions of the organization and now are adopted as generic software products by other industries. In Finland, this development has been visible in extending the scope of enterprise solutions from production to other parts of the enterprise, especially in pulp and paper manufacturing industries and in the largest companies in the metal industry.

One obvious perspective of this phenomenon is diffusion and adoption of enabling technologies, with the software technology originally embedded into or at least bundled with other technologies. Previous research has approached technology diffusion and adoption from multiple viewpoints. For example, Cooper and Zmud (1990) analyze technology diffusion within an organization from the perspective of the end-user organization (i.e., the vertical industry enterprise) based on the work of Lewin (1947) and Rogers (1983). The process starts with initiation and adoption of the technology, continues with adapting it for use in the organization, and progressing to acceptance with routinization and infusion.

Moore (1995) observes the adoption of specific technologies starting from innovators and early adopters through a product creation chasm, to the early and late majority, and finally to laggards. However, in this process, vertical markets are considered from the viewpoint of the specific technology only as early markets for the technology developer. In other words, the originally generic technologies are thought to make use of vertical market segments as stepping stones toward the main horizontal market, with the role of specific vertical businesses as hosts for future generic businesses neglected. If this were the case, IBM should never have been the mother of the primary software industry, as a vertical computer hardware manufacturing company.

Along the same lines, the life-cycle and growth models proposed by Churchill and Lewis (1983), Greiner (1972), Kazanjian (1988), McHugh (1999), Nambisan (2002), and Scott and Bruce (1987), observe the life cycle of companies in general or high-technology companies in particular from the perspective of the software company, rather than

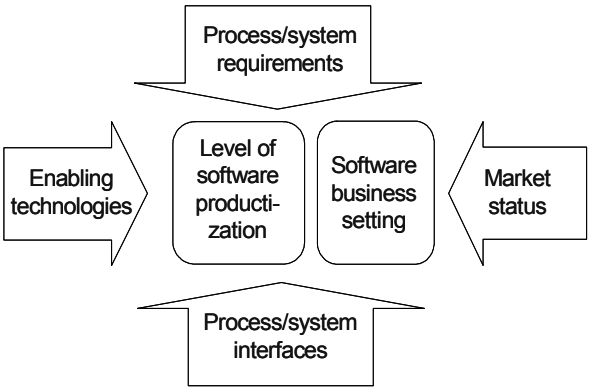

Figure 2. Factors Impacting the Success of Software Solutions in Vertical Markets

from the perspective of the status of some vertical market. However, Greiner (1972) states that the growth of a company is often related to the market environment and stage of the industry. From our perspective this is essential, since the development of software businesses servicing specific vertical markets depends both on the status of the primary software industry and the vertical market.

The key factors impacting the success of software products aimed at a vertical market can be viewed from the perspective of Figure 2. In addition to the viewpoint of the diffusion of enabling technologies (arrow from the left) and the market status (arrow from the right), these factors involve the functional requirements set forth by the processes or systems to be automated using software (arrow from above), and the interfaces through which the software interacts with these processes or systems (arrow from below). Referring to Figure 1, the level of productization and the business setting characterize the total software offering.

Considering the functional requirements, it is clear that any piece of software should be customer-oriented (McHugh 1999) in the sense of satisfying certain information processing needs. This requires expertise from the target processes or systems. Bell (1995), for example, emphasizes the importance of targeting niche markets, industry-specific conditions, and relationships with important customers in creating this expertise. In practice, the expertise may initially be realized in the form of the functionality of a unique piece of software. The level of automation of the target process or the degree of productization of the target system are often used to characterize the maturity of the solution in terms of functionality, and depend on the diffusion of the enabling technologies for which the solution has been developed.

Leaving the technology perspective aside, when an individual customer adopts a new or a modified process or system solution, a requirement to implement a new functionality emerges. When the processes or systems of a number of customers in the market settle down, it is able to productize the required solution instead of tailoring it for each individual customer. This means, however, that the software starts to depend more on the hardware and software interfaces common across the market or at least its segment. In other words, not only the required information processing functionalities and the technologies, but also system interfaces, should become standardized (Northrop 2002).

Rather than observing a vertical market from the viewpoint of particular enabling technologies or specific functional process or system requirements, we will focus on the right-hand part of Figure 2. We will, therefore, address the evolution of vertical markets as the source of software businesses, making explicit the resulting changes in business settings. In particular, our aim is to gain a better understanding of vertical market development as a basis for interaction between the secondary and primary software industries.

We need, therefore, to consider

- What is the overall change in the software markets?
- To what extent, in terms of revenues, are primary and secondary software businesses involved in the change?
- How does the change of personnel in software businesses correspond to the market change?

In this research the questions are analyzed by grounding source data to the primary and secondary software industry in Finland. The research was funded by Tekes (The National Technology Development Agency of Finland), with the aim of investigating the size and status of the national software cluster, and the potential of this cluster to expand to new markets.

3 EMPIRICAL DATA COLLECTION

Because very little empirical data has been gathered and analyzed on the secondary software industry, we will discuss in detail how the data was acquired and organized for analysis. First of all, the data collection has to take place in a relatively small country with a reasonably well-developed software industry. With this approach, data collection is feasible and the results can be generalized to some extent to represent the software industry evolution globally due to the use of global industry standards and platforms for products and services (Gawer and Cusumano 2002; Hoch et al. 2000). Previous research results related to software firms originating from Canada, Finland, Ireland, New Zealand, Norway, Malaysia, and the United States imply that software businesses are using similar strategies in their internationalization and business processes (Bell 1995; Coviello and Munro 1997). We chose Finland with its 5.2 million inhabitants as the target of our study based on relatively good access to empirical data compared to the other alternatives.

The first goal of the empirical data collection was to estimate the number of software business in Finland in terms of the annual sales volume and the number of software development professionals. For the purpose of this study, *software business* was defined as business operations based on software being developed by the organization. This includes business operations within enterprises classified as software enterprises under category 72.2, as well as similar business operations in software-oriented business units in the secondary software industry and in public administration.

Furthermore, professionals with software development competence were included also from the public sector as well, although they do not work for business organizations, while the personnel supporting the operational use of ICT systems was mostly excluded.

The rationale was that their actual software development competence could not be evaluated, thus causing problems in data collection. This definition also excluded non-software activities performed in category 72.2 companies, such as language translation and logistics consulting.

The focus was on the largest companies in Finland and on enterprises classified as software companies. The 250 largest companies measured by revenue were chosen as the target group. In addition, 71 companies ranked to positions from 251 to 500 by revenue were included in the target group, considering their potential for software business that was evaluated based on public information about their activities.

The number of employees in these 321 enterprises was about 910,000 in 2002 and the total revenue was 246,000 M€. The smallest revenue made by a company included in the data was 56 M€, and the smallest companies in terms of the personnel volume had fewer than 300 employees. Assuming these companies operate on verticals, which spend typically less than 2 percent on ICT, this would imply about 1 M€ ICT annual spending and much below 10 professionals in a single vertical enterprise. As a comparison, the employee union of software professionals has a registry of about 6,000 software service companies (with over 5,000 having some revenue), out of which 425 registered software companies had annual revenue above 1 M€ (i.e., the annual ICT budget mentioned above).

The list of 321 large companies with potential internal software business units included 16 software companies. This list was merged with a list of 348 smaller software companies, adding up to a target list of about 670 companies. Some small software companies whose basic data was not available were dropped from the target list and some individual companies with a software business were added during the data collection process.

The data was produced through two methods. We used *company-specific data* to analyze the companies, this information being available through a questionnaire, annual reports, web, or other sources. The companies were clustered according to a classification of 24 vertical markets. For each market we produced *vertical market estimates* based on the company-specific data available from the companies in that vertical market, the vertical market interviews, and other public data. These vertical market estimates include average spending on internal software development, average proportion of software professionals from the total number employees, among others. In a later phase, these vertical market estimates and the overall volume data of specific companies were used to produce estimates for companies whose data was not available. Totals were calculated by adding up the company-specific data or estimate for each of the 675 companies. Final figures include these totals, the estimated activity of companies excluded from the data collection process, as well as the software production data from the public administration.

Company-specific data was collected first by questionnaires delivered on paper and as web questionnaires. Each target company received a form for the enterprise and a form to be delivered to each business unit that produces software within the enterprise. Responses were received from 75 companies, out of which 42 were software companies. Out of the 33 responses from the vertical market enterprises, 31 were within the 250 largest companies. The percentage of the responses was considered insufficient for producing accurate estimates, and further company-specific data collection was included

in the vertical market interviews. Therefore, in the second phase, the 24 vertical markets were analyzed using structured interviews.

A total of 52 interviews were conducted using phone conversations lasting from 10 minutes to an hour. The respondents were typically CIOs, CTOs, or ICT managers of the largest companies or managing directors of software companies. Typically one to two top companies from each vertical market were interviewed, with the aim of finding the person most knowledgeable about software development and utilization status at the specific vertical industry. The interviewed companies included the 10 top companies when measured with research and development spending, the largest software houses, the software trade unions, and the key public software organizations.

The interview themes included the typical spending on software development and ICT on enterprises in that vertical industry and company-specific figures for the largest players, characteristic features of software development in the verticals, size of software development organizations, existing and estimated appearance of spin-off companies, use of software services and products, outsourcing, potential for international software business based on vertical market software, and free comments. The interviews produced company-specific data for 60 software-intensive enterprises. When added to other data from 38 companies and the 75 interview results, 173 company-specific estimates were made. These results were combined with the vertical market estimates from the interviews in order to produce the final vertical market estimates. In addition, the interviews provided qualitative data.

4 CHANGES IN MARKETS AND PERSONNEL IN THE PERIOD 2000–2003

4.1 Software Competence in Finland

Based on the data gathered, an overview of the software industry in Finland is presented in Figure 3 from the viewpoint of the primary and secondary software industries (SWI) and the contemporary locations of software development professionals. The primary software industry employed about 30,000 software professionals in 2003 and somewhat less in the secondary software industry, adding up to about 60,000 employees in total. Out of these, between 11,000 and 14,000 employees worked on software products, which is defined as sales of copies of software with a license to use it. About 5,000 of the people involved in the software product business were employed in the secondary software industry, such as software business units of telecommunication and automation companies.

4.2 The Software Market

The primary software industry generated over 3,000 M€ annual revenue related to software, while the secondary software industry was estimated to produce only about

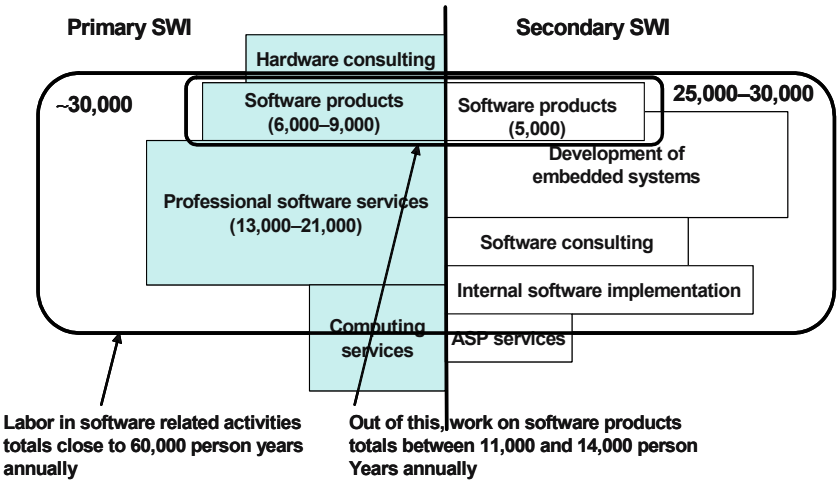

Figure 3. Volume of Software Businesses Within Units Located in Finland in 2003 (boxes with rounded corners; shadowed boxes with squared corners represent the sub-categories of enterprises, corresponding to NICE 72.2)

1,000 M€ revenue with almost the same number of employees (Figure 4). Out of the total over 4,000 M€ software revenue, roughly 1,100 M€ to 1,400 M€ was generated from software products. From this, less than one half (500+ M€) was estimated to be generated from the products of the secondary software industry. The interviews implied that this was due to the habit of vertical market enterprises to sell solutions that include software so that they are adding the software development cost to the product price, when selling embedded systems, or to the customer service price, when offering ASP services.

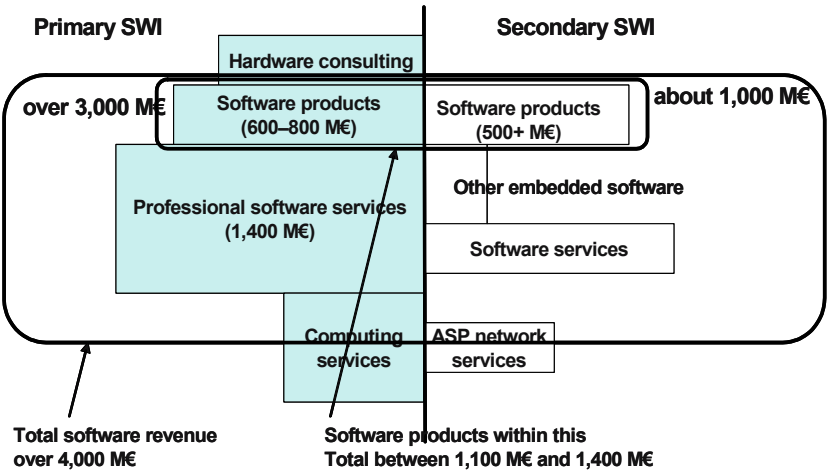

Figure 4. Total Volume of Software Business in Finland in 2003

When comparing these figures with other statistical material, one should notice that the statistics typically include only sales (including non-software sales) generated by the verticals or only the product or service sales of the primary software industry. Other sources also count only the revenue of companies with headquarters in Finland, while these estimates include revenue and head-count of business units located in Finland regardless the ownership of the company and the industry classification.

4.3 Relocations of Professionals

Figure 5 represents personnel changes between the primary and secondary software industries in Finland from early 2000 to late 2003, indicating also the types of businesses shown in Figures 3 and 4. During this period, 2,000 software product development professionals were estimated to have moved from the secondary software industry to the primary industry's small software product companies, as individual employees or along with spin-offs of entire software product businesses from the secondary industry's verticals.

Moreover, existing large software service enterprises of the primary industry acquired smaller software product companies, adding up to a total of 2,500 employees shifting from the small software ventures to the software product business units of these enterprises. Typically, this took place when the small software venture reached the size of about 20 to 30 employees, and turned profitable. In the later part of the period about 1,500 employees returned to the small software ventures from the big enterprises, for various reasons including management buy-outs. As these figures indicate, the turn-over of personnel has been rather hectic in small product ventures during the period of observation, in addition to the growing trend of personnel.

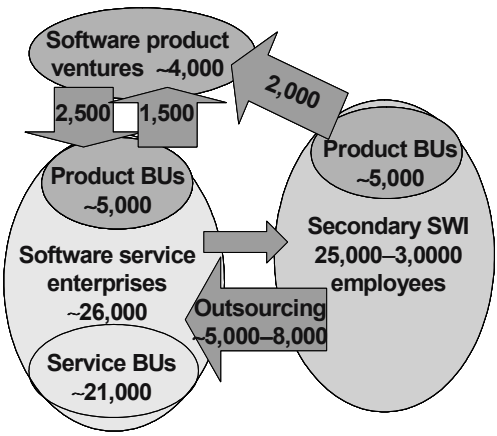

Figure 5. Employee Volumes During 2003 and the Main Employee Transitions in Finnish Software Businesses During 2003–2003 (Software product ventures refers to independent, typically small, product companies; product BUs refer to product business units within the primary [on the left] or the secondary [on the right] industry)

The number of software professionals outsourced from vertical industries to the primary software industry during this period includes between 5,000 and 8,000 employees, while little insourcing took place during the period. When the statistics of the primary software industry are checked against this data, we can see that the software service business matches well with the figures of outsourcing with little organic growth in the services business, while the software product business grew faster during the period of observation. Moreover, we also notice that, early in 2000, the secondary software business actually employed more software development professionals than the primary software business.

5 ANALYSIS OF THE INTERACTION OF SECONDARY AND PRIMARY SOFTWARE INDUSTRIES

Based on the empirical data, the interaction of the primary software and secondary software industries seems to follow a common pattern, where software development initiated within a vertical enterprise follows the life-cycle presented in Figure 6. In the first phase (the *innovation phase*), the enterprise invests increasingly in ICT, including internal software development, but cuts back the software workforce during economic downturns by relocating them to other duties. In this phase, some of the software development professionals become self-employed by establishing micro ventures to capitalize their expertise across vertical industries.

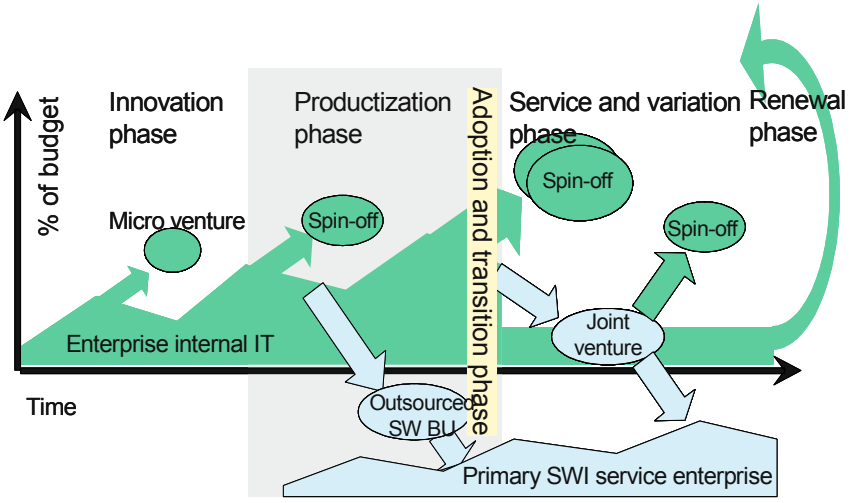

Figure 6. Life-Cycle of Software Development in a Vertical Industry Enterprise (the horizontal axis represents time and the vertical axis spending on ICT and software, which represents approximately the percentage of internal software professionals in the enterprise)

In the second phase (the *productization phase*) the percentage and volume of internal software development has risen, often as a consequence of the implementation of enterprise-specific information systems. After accomplishing such assignment, however, the whole ICT department may be outsourced during a recession, to form a separate software service business unit. Sometimes the ICT department may also spin off as a distinct software venture. During this phase, the ICT supported core processes of enterprises within the vertical market are typically being harmonized, and the operational environments and the required hardware and software interfaces are becoming standardized. The effort needed to implement a software product that stems from the evolution of a vertical market, or secondary software business, depends heavily on degree of standardization of hardware and software interfaces (Garud and Kumaraswamy 1993) as well as on the standardization of the processes. This sets the minimum for organization size sufficient for productizing the offering, leaving the majority of the small software houses out of the new market niche due to major shortcomings in their competence portfolio.

In this phase, one or some of the spin-off ventures or occasionally an existing primary software venture focusing on the emerging market has been able to create the first software products matching with the unifying functional requirements of the core processes and standardizing interfaces. In the next phase (the *adoption and transition phase*, vertical bar in Figure 6), at least some of the leading companies, if not most companies, adopt this solution. As one of the consequences, they may outsource a majority of their internal software personnel. If the number of the outsourced employees is in the range of 50 or more professionals, the enterprise can form a joint venture with a primary industry's software service house. Typically this involves a few years' agreement on maintaining the existing core operational systems of the enterprise, or an assignment to help with emerging common software solutions. During the period of observation, this was the usual approach, especially in the banking industry and in specific segments of the electronics and manufacturing industries.

Software departments with fewer than 50 employees are typically simply outsourced or asked to create a software spin-off company. The reason is that if a smaller number of software professionals have been sufficient for the vertical company to automate its main operational processes or to start deploying software in its products, it wants to focus on its core business and competences. A spin-off company or a software outsourcing partner often starts selling its services or products to competitors of the previous host company. In the innovation phase or in the early productization phase this would not have been possible, because software was considered as a strategic asset. In the adoption and subsequent phases, software is considered mainly as a commodity product (i.e., a cost to be minimized).

After the market-share battle of the adoption and transition phase has ceased, the secondary software market falls to the *service and variation phase*, where one or a few software product vendors dominate the product market, and internal software development resources have mostly or entirely been replaced by professional services purchased from the primary industry's companies. The ICT spending, which in the standardization phase was commonly over 5 percent and as much as 10 percent or more in some vertical markets (for example, retail banking and telecommunication services in our source data), falls in the service and variation phase verticals to a level of 1.5 to 2 percent (for example, in the transportation, furniture, basic metal, textile, and retail industries), relying mainly on external software and services.

The vertical markets in which companies follow this pattern of developments are defined by their major information processing needs. Automating the major process with software can be essential for reducing cost as it may account for 50 to 85 percentof all communications in an organization (Tyrväinen et al. 2005). But if the in-house software competence is outsourced when adopting an external product for the main process in the adoption phase, there is little capability to innovate new software supported business models for the enterprise. If a company that is in the middle of the service and variation phase changes its business model in such a way that it requires a new major information processing system or an innovative software solution for its product, not available in the market place, it enters a *renewal phase*. In this phase, the company has two options: either to use external partners to implement the solution or to create and keep the required knowledge in-house by developing the solution with its own personnel. In the latter case, the company creates a new vertical (sub)market and shifts back to the innovation phase, such as Merita bank establishing an Internet banking vertical in Finland in the 1990s or Wall-Mart stores in the retail industry in the United States.

6 IMPLICATIONS FOR SOFTWARE BUSINESS STRATEGIES

6.1 Drivers for Change

The drivers for change in a vertical software industry are presented in Table 1 together with the characteristic features related to the business strategies applicable in each phase. In the data we collected, the vertical industry enterprises driving the adoption of a new technology to the industry prefer using their internal processes in order to maintain their competitive advantage based on process differentiation or improved cost performance in the innovation phase. After a while, competitors are able to copy the best practices adopted in the industry and start asking for software products from the primary software industry ventures in the productization phase.

From the technology viewpoint, the innovators developing new technology try to secure it with patents and use proprietary closed interfaces for software development. The leading hardware vendors are typically reluctant to share their knowledge with software companies, creating entry barriers for competitors. They implement the software needed for operating their systems internally, creating unified hardware/software systems. In this phase, the software business is limited to company internal or external software services. Independent software houses cannot compete with a single vendor who owns the technology platform needed for operating the software. In case such software appears, the proprietary interface can be changed or the platform vendor can bundle free software with the hardware. In case the market is not large and attractive enough, the software market evolution may stop with the innovation phase. Typical examples from the Finnish software market stuck at the innovation phase include markets with globally operating strong companies, such as the elevator software market dominated by Kone Corp. and weather measurement sensor software dominated by Vaisala Group.

Table 1. Changes in Vertical Industry Processes and Available Technology Set the Pace for Vertical Software Industry Evolution

Drivers and Features	Innovation Phase	Productization Phase	Transition Phase	Service and Variation Phase
Process Drivers (Vertical Industry)	Process innovation, competitive advantage (differentiation/costs), enterprise-specific processes	SW supported processes, process harmonization, repeatability between enterprises	Process standards adopted	Process optimization, IT process cost optimization
Software Business	Customer-specific services, integrated HW/SW products, an application per enterprise	Productization, competing standards, micro ecologies, network development	Market share race, competing ecologies	New development on the dominant SW platform, open source
Technology Drivers	Technology innovation, competing with IPRs and closed interfaces, equipment vendors dominating SW market	Technology standardization within vertical market segment, multiple strong vendors	Fast technology adoption, market expansion	Technology dominance, bulk hardware with standard interface and large user base

The early console game vendors operated in the innovation phase markets with a similar strategy, competing primarily with new technology and platforms while maintaining the process of playing and game software application development in-house. However, after some years the platform was no longer the only criteria for choosing between competing game offerings, the game markets required more effort to develop software applications and forced this vertical industry to shift to the productization phase. This has enabled an increase of game software developers in independent game companies, improved offerings to customers, and the increased industry market volume. Increasing market volume also attracts more vendors, creating more competition and enabling specialization. These further lower the prices, attracting more customers.

When the productized offerings satisfy the requirements of the processes of the vertical industry companies, the market enters the transition phase. Typically, one of the competing ecologies or standards reaches a dominating position with a better offering creating a dominant design (see Murmann and Frenken 2006) in the service and variation phase while some alternative designs may be better applicable for specialized purposes, such as the Macintosh computer platform, long preferred by media processing industries. Although open source software development appears in the productization phase, it is mostly visible in the service and variation phase.

The industry evolution may also stop at the productization phase, if none of the competing technologies and interfaces gains sufficient critical mass to drive the positive cycle. For example, there has been a wide variety of field busses used by various industries, none of which has been accepted widely. Some of them have better cost structure for industries with easier requirements, while the redundant fast field busses are too expensive or less applicable for others.

6.2 Importance of Interfaces

Interfaces are commonly used as a means to modularize systems. The effort needed to implement a software system can be minimized by using existing software imple-

mented by others through the interfaces. The existing systems act as the vessels for transferring knowledge from other organizations to the software systems developer. Thus availability of system components with easy to use interfaces greatly reduces the knowledge required from the software developers. If the interface is standardized, the software developer does not have to understand the details of the system used, while nonstandard interfaces with variations require developers to have more knowledge of the system used. The requirement to master a wide number of proprietary technologies as well as the requirements of the industry process knowledge can be too much for most software development organizations. This can limit the critical mass of organization needed for reaching the next phase in the vertical industry evolution.

Consider, for example, the software organization needed for implementing a basic retail banking system (Table 2). The system needs to interface with the customers, needs to deliver notes, needs to communicate with a teller, needs a mainframe computer, and a database management system to the server. The interfaces available for these technologies were long proprietary, as listed by configuration A in Table 2, limiting the number of vendors to a few. Instead, after evolution of the technologies, the interfaces needed were standardized, making it easy for a larger number of companies to implement the system.

From the viewpoint of the interfaces, the large IT service companies with access to a broad competence pool tend to dominate the innovation phase markets. Smaller product companies appear more often in the productization phase where emerging standards can be utilized. In the last phase, the dominating design again favors large companies. Note also that a small company adopting configuration B in the example has dropped the "note delivery" interface from the system offering, which is based on use of common Internet technologies. This means that this inexpensive offering can be used by retail banks willing to drop this functionality from their offering or able to outsource it. In other words, the development of the set of technologies needed by a vertical industry (retail banking) impacts the cost/functionality of the software offering available to the vertical industry companies, but the software offering can also impact the business model of the industry by creating a market for ATM outsourcing.

Table 2. Alternative Interfaces Used by a Retail Banking Application (The interfaces on the left are mainly from the innovation phase and require high level of competence available only in a few software enterprises or in-house; the interfaces on the right are from later date and require only knowledge available for millions of software developers)

Hardware interface	A – Early Configuration		B – Late Configuration	
User Interface	ATM hardware interface	Proprietary	Browser interface (SSH, HTML, TCP/IP)	Standard
Note Delivery	ATM message interface	Proprietary	–	–
Clark Interface	Windows workstation	Standard	Browser	Standard
Server Computer	IBM Mainframe	Proprietary	Linux / Windows	Standard
Database System	IBM DB2	Proprietary	MySQL	Standard

6.3 Shift of Software Development

One of the key findings of this research is that the focus of a vertical market shifts inevitably during the market evolution from the use of internal resources to extensive use of software products and professional software services provided by the primary software industry. Figure 7 represents this by dividing the IT spending on software related activities to internal software development, external software services, and software products (the trumpet model). In the innovation phase, a major part of the activity takes place within the enterprise while the share of software products and services increases along with development. Some verticals may remain in the early phases for years, such as several manufacturing segments where sufficient volume for software productization does not exist, while most of the verticals move forward. Thus this development is visible also in the total figures of software spending. According to EITO (2004), 52 to 59 percent of IT spending in Europe, Japan, and the United States was on internal software development in 1993, while purchased services and software packages were 14 to 32 percent and 10 to 29 percent, respectively. In our source data collected 10 years later, these three segments were roughly equal, indicating a major drop in internal spending on

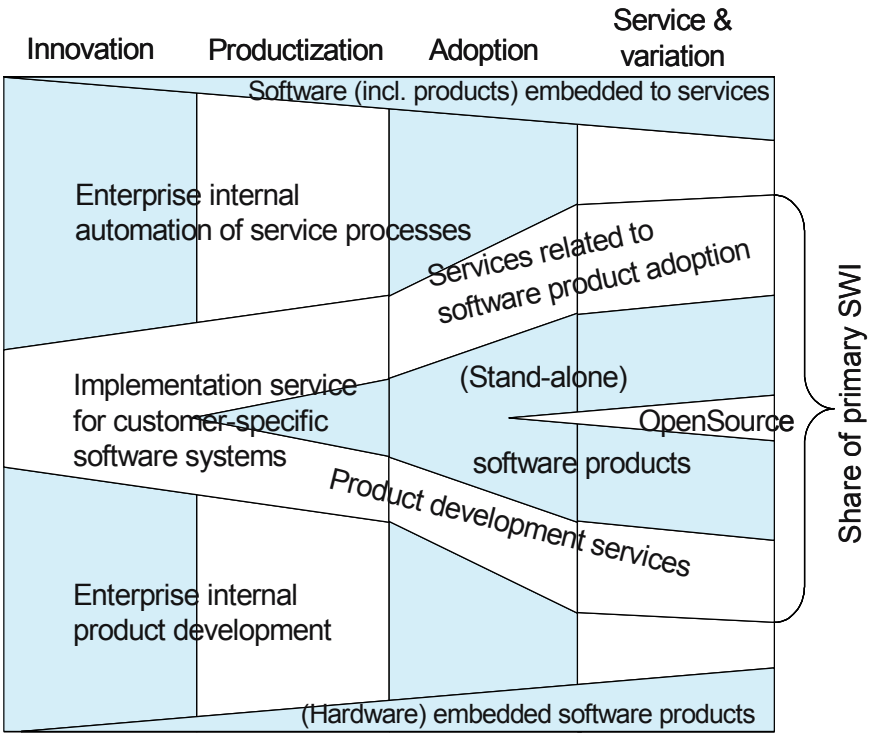

Figure 7. Share of Software Development Activity in Each Phase of the Vertical Market Change

behalf of purchased services and software products. Similar figures (33 percent, 35 percent, and 32 percent, respectively) were also produced by a commercial market study of Finnish software markets (Market-Visio 2003). There are still industries where large vertical industry enterprises want to maintain strong control over the innovative software-supported services and keep the majority of software development in-house, such as the telecom operators (see Dittberner Associates 2004).

The Finnish data indicates, in terms of the volume of software professionals, including software product developers, that the primary and secondary industries are much alike. In terms of visible revenues, the secondary software industry is considerably worse, as the added value of the software is sold along with a hardware product (e.g., a mobile phone) or in the price of a service (e.g., banking service fees). The evolution may also stop in the early phases if the market size is insufficient to attract enough competition to the segment or the dominating companies are unwilling to proceed. Some dominating players in specific market niches in the manufacturing industries are especially unwilling to open their software interfaces to open competition. Therefore, the main implication from a strategic perspective is when to initiate and how to implement responses to the market change in order to create profitable software businesses.

6.4 Software Business Opportunities in Each Phase

Because the vertical market evolution affects software business opportunities, it is useful to make an overview of the means to cope with the change during the main phases of the evolution. In the innovation phase, the software developed is customer-specific and no place exists for a profitable product business. However, software service companies can evaluate and learn new technologies and customer processes from the leading vertical market companies.

In the productization phase, the implementations made in the leading companies are used as templates or specifications for implementations in other companies. In this phase, software companies can make use of product platform strategies while standardizing their offerings to the vertical. Standardization of functionality and equipment enables division of work by hiding functionality behind proprietary or standard interfaces. As soon as all the interfaces and the application processes are standardized, the vertical market is willing to adopt a total solution serving the core business processes of the vertical. The companies involved in standardizing the key interfaces and customer processes gain the main advantage in this competition. Companies involved in standardizing a new core interface form a micro ecology, providing a strong total offering for the vertical market. This attracts new technology and software providers as well as customers willing to adopt a productized package. To get positioned well in the networks developed in the productization phase, it is important for the software houses to serve at least one of the leading customers of the vertical market and to adopt the standard interfaces or participate in their development.

For example, Digia was a small local software company until it joined the Symbian consortia, which was developing an operating system platform for intelligent mobile phones. With the aid of the micro ecology around the Symbian development, Digia was able to harness the growing use of Symbian products and grow rapidly to one of the large national software houses with a strong international customer base in the telecommuni-

cations industry. Altogether, integrating into emerging micro ecologies in the productization phase seems to be the most effective means to grow in the vertical industry software business.

In the adoption phase, the networks providing offerings for the vertical market compete with each other. A majority of the companies adopt a software package, outsource the majority of their in-house software development, and lower their IT spending. This provides an opportunity for the software service providers to extend their business. For the software product houses, this phase means competition for the critical market share. In some cases there is only a single offering, which easily dominates the market, causing a fast move to the next phase. In any case the market share battles of multiple networks or companies do not last long compared to the lengthy development in the previous phases.

In the service and variation phase, the vertical market has adopted a dominant solution. A major part of IT spending goes to software products and external services. The dominant product vendor provides little market opportunity for other software products while the service companies may deploy a customer intimacy strategy. Other software product companies may try to use disruptive innovation to evolve the vertical into a renewal phase, where the vertical market companies need to reposition themselves into a new market setting, insourcing software development or making use of external services and products to do so.

7 SUMMARY AND FURTHER WORK

This paper presented a model for evolution of the vertical software industry based on data collected from Finnish secondary and primary software industries. The model describes how industry-specific process knowledge and technology are combined with tailored software to create competitive advantage to the industry enterprise. Due to competition, the business processes tend to harmonize and technologies tend to standardize, enabling productization of the software by micro ecologies of companies. Later on, some of the solutions may become the dominant design, serving as a platform for new innovations.

The model presented here adds to the literature by taking the viewpoint of an industry with enterprises having similar dominant processes and internal software development organizations. Innovation diffusion models (e.g., Rogers 1983) tend to address the impact of single technologies on multiple industries and do not address the shift of knowledgeable software personnel from vertical industries to the primary software industry nor the changes in investments on technology development within the vertical industry. The model presented here carries some similarities with the industry life cycle theory (Gort and Klepper 1982), which predicts that the number of companies in a new product industry will grow until a shake down reduces it to a lower, stable level. Based on the data we have collected, the evolution of a secondary software industry may proceed through all of the phases presented here, producing results similar to the ones in the industry life cycle theory. However, the development may stop for a while at a phase, causing a shakeout to take place and a new take-off to appear after it. Also, neither the technology adoption model nor the industry life cycle theory addresses the shift of

personnel between the secondary and the primary software industries, the interrelationship of standardizing technology with the process of harmonization, and software market evolution as presented here.

Although the analysis presented in this paper is based on a single geographical market that represents only a small fraction on the global market, the results can be generalized with some caution to represent software industry evolution globally due to the use of global industry standards and platforms for products and services. Market statistics indicate that the vertical markets in different countries can be in different phases of development prior to reaching the final phase, usually following the development of the leading customer companies of the vertical market in the geographical area, or due to the specific technical innovations being developed.

Clearly, the software companies basing their operations in markets progressing ahead of the same vertical markets have a competitive advantage while internationalizing their business. Especially companies that establish or join new micro ecologies around new standards in the productization phase will have a clear advantage when the markets grow and they can later capitalize on their dominant design position in the mainstream market. On this basis, the Technology Agency of Finland (Tekes) initiated a further research action to analyze six chosen vertical software markets, where the potential of Finnish software companies was considered high based on their life-cycle phase. These industries were the finance, construction and real estate, forest, energy, machine manufacturing, and retail/wholesales industries. In addition, a separate project, SmarTop, was started to analyze the software market of telecom operator products (see Mazhelis et al. 2007). The results of these later studies elaborate further the phenomena, especially the role of alternative actors in the evolution as well as the impact of multiple core processes on the vertical industry company, but are beyond the scope of this paper.

Acknowledgments

The authors acknowledge gratefully the financial and other support given for the study by Tekes, especially Mr. Matti Sihto, and the other members of the steering group of the research project.

References

Bell, J. 1995. "The Internationalization of Small Computer Software Firms. A Further Challenge to 'Stage' Theories," *European Journal of Marketing* (29:8), pp. 60-75.

BMBF. 2000. *Analyze und Evaluation der Softwareentwicklung in Deutschland*, Bundesministerium für Bildung und Forschung (Federal Ministry of Education and Research), Germany.

Campbell-Kelly, M. 2004. *From Airline Reservations to Sonic the Hedgehog*, Cambridge, MA: MIT Press.

Chesbrough, H., and Rosenbloom, R. 2002. "The Role of the Business Model in Capturing Value from Innovation: Evidence from Xerox Corporation's Technology Spin-Off Companies," *Industrial and Corporate Change* (11:3), pp. 529-555.

Churchill, N. C., and Lewis, V. L. 1986. "The Five Stages of Small Business Growth," *Harvard Business Review* (61:3), pp. 30-49.

Colecchia, A., Anton-Zabalza, E., Devlin, A., and Montagnier, P. 2002. *Measuring the Information Economy*, Paris: Organisation for Economic Co-operation and Development, Directorate for Science, Technology and Industry.

Cooper, R., and Zmud, R. 1990. "Information Technology Implementation Research: A Technological Diffusion Approach," *Management Science* (36:2), pp. 123-139.

Coviello, N., and Munro, H. 1997. "Network Relationships and the Internationalization Process of Small Software Firms," *International Business Review* (6:4), pp. 361-386.

Dittberner Associates. 2004. "Dittberner's OSS/BSS KnowledgeBase™," December (http://www.dittberner.com/reports/about53.php).

EITO. 2004. *European Information Technology Observatory*, European Information Technology Observatory Portal.

Gawer, A., and Cusumano, M. A. 2002. *Platform Leadership: How Intel, Microsoft, and Cisco Drive Industry Innovation*, Boston: Harvard Business School Press.

Garud, R., and Kumaraswamy, A. 1993. "Changing Competitive Dynamics in Network Industries: an Exploration of Sun Microsystems' Open Systems Strategy," *Strategic Management Journal* (13:5), pp. 351-369.

Giarratana, M. 2004. "The Birth of a New Industry: Entry by Start-Ups and the Drivers of Firm Growth the Case of Encryption Software," *Research Policy* (33), pp. 33, 787-806.

Gort, M., and Klepper, S. 1982. "Time Paths in the Diffusion of Product Innovations," *Economic Journal* (92), pp. 630-653.

Greiner, L. E. 1972. "Evolution and Revolution as Organizations Grow," *Harvard Business Review* (50:4), pp. 37-46.

Hoch, D. J., Roeding, C. R., Purkert, G., Lindner, S. K., and Müller, R. 2000. *Secrets of Software Success*, Boston: Harvard Business School Press.

Kazanjian, R. K. 1988. "Relation of Dominant Problems to Stages of Growth in Technology-Based New Ventures," *Academy of Management Journal* (31:2), pp. 257-279.

Kuitunen, H., Maula, M., and Kontio, J. 2005. *Software Product Industry Survey 2005, Finnish Software Product Business in 2004*, Espoo, Finland: Helsinki University of Technology.

Lewin, K. 1947. "Group Decision and Social Change" in *Readings in Social Psychology*, T. M. Newcomb and E. L. Hartley (eds.), New York: Henry Holt.

Mazhelis, O., Tyrväinen, P., and Viitala, E. 2007. "Modeling Software Integration Scenarios for Telecommunications Operations Software Vendors," in *Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management*, M. Helander, M. Xie, R. Jiao, and K. C. Tan (eds.), Los Alamitos, CA: IEEE Computer Society Press, pp. 49-54.

Market-Visio. 2003. *IT-investoinnit, -kustannukset ja –hankkeet Suomessa 2003-2005*, Helsinki, Finland: Market-Visio ICT Research and Advisory Services.

McHugh, P. 1999. *Making it Big in Software*, Triverton, UK: Rubiv Publishing.

Moore, G. A. 1995. *Inside the Tornado: Marketing Strategies from Silicon Valley's Cutting Edge*, New York: HarperCollins Publishers.

Murmann, J. P., and Frenken, K. 2006. "Toward a Systematic Framework for Research on Dominant Designs, Technological Innovations, and Industrial Change," *Research Policy* (35:7), pp. 925-952.

Nambisan, S. 2002. "Software Firm Evolution and Innovation-Orientation," *Journal of Engineering and Technology Management* (19), pp. 141-165.

Northrop, L. 2002. "SEI's Software Product Line Tenets," *IEEE Software* (19:4), pp. 32-40.

Rogers, E. M. 1983. *Diffusion of Innovations* (3rd ed.), New York: The Free Press.

Romijn, H., and Albaladejo, M. 2002. "Determinants of Innovation Capability in Small Electronics and Software Firms in Southeast England," *Research Policy* (31), pp. 1053-1067.

Scott, M., and Bruce, R. 1987. "Five Stages of Growth in Small Businesses," *Long RTange Planning* (20:2), pp. 45-52.

Tyrväinen, P., Kilpeläinen, T., and Järvenpää, M. 2005. "Patterns and Measures of Digitalization in Business Unit Communication," *International Journal of Business Information Systems* (1:1/2), pp. 199-219.

About the Authors

Dr. **Pasi Tyrväinen** is Professor of Computer Science and Information Systems (Digital Media) at the University of Jyväskylä. He received his doctorate from Helsinki University of Technology in 1994. Professor Tyrväinen's research focuses on software business, digital rights management, and enterprise content management. In addition to his nine years at the University of Jyväskylä, he has twelve years of prior experience in industry as an R&D director at Honeywell Industrial Control and in the software laboratory of Nokia Research Center. He also supports the National Technology Foundation (Tekes) in directing government R&D activities related to the software industry. He has over 100 scientific and professional publications, of which about 60 appear in scientific forums. He can be reached at pasi.tyrvainen@jyu.fi.

Professor **Juhani Warsta**, Ph.D., currently works as an acting professor of software business at the Department of Information Processing Science, University of Oulu, Finland. He has several years work experience in different sectors of the software business, both from the production perspective as well as the customer perspective. He is actively involved in software business research as well as e- commerce aspects of business. He can be reached at juhani.warsta@oulu.fi.

Dr. Econ, Dr. Tech **Veikko Seppänen** is the Deputy CTO of Elektrobit Corporation Plc., a global wireless systems and automotive software company. He is also a part-time professor of software business at the Department of Information Processing Science, University of Oulu, Finland. He has defended two doctoral theses, the first on software reuse in 1990 and the second on industrial marketing in 2000. Before joining Elektrobit in 2002, he held various research and management positions at the Technical Research Centre of Finland (VTT) from 1983 to 2000 and worked as a full-time software business professor at the University of Oulu from 2000 to 2002. In 1986-1987, he was a non-degree graduate student and Asla Fulbright Scholar at the University of California at Irvine, and in 1991-1993, was a JSPS Postdoctoral Fellow at the University of Kyoto, Japan. Dr. Seppänen has authored over 100 scientific and professional publications. He can be reached atveikko.seppanen@oulu.fi.